Robust Tensor Completion via Capped Frobenius Norm

Xiao Peng Li, Zhi-Yong Wang, Zhang-Lei Shi, Hing Cheung So, Fellow, IEEE, and Nicholas D. Sidiropoulos, Fellow, IEEE

Abstract—Tensor completion (TC) refers to restoring the missing entries in a given tensor by making use of the low-rank structure. Most existing algorithms have excellent performance in Gaussian noise or impulsive noise scenarios. Generally speaking, the Frobenius-norm-based methods achieve excellent performance in additive Gaussian noise, while their recovery severely degrades in impulsive noise. Although the algorithms using the $\ell_p$-norm ($0 < p < 2$) or its variants can attain high restoration accuracy in the presence of gross errors, they are inferior to the Frobenius-norm-based methods when the noise is Gaussian-distributed. Therefore, an approach that is able to perform well in both Gaussian noise and impulsive noise is desired. In this work, we use a capped Frobenius norm to restrain outliers, which corresponds to a form of the truncated least-squares loss function. The upper bound of our capped Frobenius norm is automatically updated using normalized median absolute deviation during iterations. Therefore, it achieves better performance than the $\ell_p$-norm with outlier-contaminated observations and attains comparable accuracy to the Frobenius norm without tuning parameter in Gaussian noise. We then adopt the half-quadratic theory to convert the nonconvex problem into a tractable multivariable problem, that is, convex optimization with respect to (w.r.t.) each individual variable. To address the resultant task, we exploit the proximal block coordinate descent (PBDC) method and then establish the convergence of the suggested algorithm. Specifically, the objective function value is guaranteed to be convergent while the variable sequence has a subsequence converging to a critical point. Experimental results based on real-world images and videos exhibit the superiority of the devised approach over several state-of-the-art algorithms in terms of recovery performance. MATLAB code is available at https://github.com/Li-X-P/Code-of-Robust-Tensor-Completion.

Index Terms—Capped Frobenius norm, proximal block coordinate descent, robust recovery, tensor completion (TC), tensor ring.

I. INTRODUCTION

TENSOR (a.k.a. multiway array) is the multidimensional extension of scalar, vector, and matrix [1] and can represent many real-world signals, including color images, videos, hyperspectral images, and radar data to name a few. Although tensors can be unfolded into matrices and then be processed by matrix techniques [2], [3], this procedure may discard the inherent information in the high order. Therefore, multilinear algebra for tensors is a powerful tool to analyze high-order data.

The success of low-rank matrix recovery [4], [5] has inspired a large number of researchers to expand the concept to tensor completion (TC) for processing high-order data. TC aims at recovering the missing entries of a partially observed tensor using low-rank structure and has a wide range of applications, such as image inpainting [6], video restoration [7], multiple-input multiple-output (MIMO) radar localization [8], object detection [9], background–foreground separation [10], [11], and hyperspectral image recovery [12]. This is because the dominant information of high-dimensional data is contained in its low-dimensional subspace.

Analogous to matrix completion, TC is intuitively formulated as a rank minimization problem with the constraint that the recovered and given tensors are identical in the observation set. Since it is difficult to handle the rank minimization problem, most existing algorithms adopt the low-rank factorization strategy to achieve TC. Note that various tensor decomposition models generate different TC methods. For instance, CANDECOMP/PARAFAC (CP) decomposition factorizes a tensor into a sum of outer products of vectors [13], [14], which has been applied to tensor restoration [15], [16], [17]. The Tucker decomposition uses one small core tensor and a set of matrices [18], and the corresponding TC algorithms include [19], [20]. Besides, tensor singular value decomposition (t-SVD) [21] has been adopted for tensor recovery [22], [23], [24]. In addition, tensor train factorization decomposes an $N$th-order tensor into two matrices and $(N - 2)$ third-order tensors [25]. It has also been exploited for TC, and it demonstrates better recovery performance than the CP decomposition, Tucker decomposition, and t-SVD [7], [26], [27]. As an improvement of the tensor train factorization, tensor ring decomposition factorizes an $N$th-order tensor into $N$ third-order tensors [28] and has been applied to tensor recovery [29], [30], [31]. On the other hand, a fully connected tensor network decomposition has been developed to restore higher order tensors [32]. Compared with tensor train and tensor ring
 factorizations, it decomposes an Nth-order tensor into N Nth-order tensors, which results in a more adequate and flexible representation.

Despite the fact that Gaussian distribution is the most typical noise model, non-Gaussian-distributed noise also appears in flexible representation. To restrain anomalies, the ℓp-norm with p ∈ (0, 2) has been applied to minimize the recovered error, resulting in iteratively reweighted t-SVD (IR-t-SVD) [36], ℓp-regression tensor train completion (ℓp-TTC) [37], and trilinear alternating least absolute error regression (TALAE) [38]. Nevertheless, the ℓp-norm-based methods have relatively high computational complexity since solving the ℓp-norm-based problem requires a multilayer iterative procedure. Besides, the ℓp-norm with p ∈ (0, 1) poses a challenge because it is a nonconvex and nonsmooth function. To tackle this issue, Chen et al. [39] proposed a logarithmic norm to approximate the ℓp-norm with p ∈ (0, 1) and apply it for tensor recovery in the presence of gross errors, yielding an algorithm called logarithmic norm minimization and outlier projection (LNOP). Li and So [40] suggest approximating the ℓp-norm with the ℓp,ε-norm, which demonstrates better performance than the ℓp-norm in robust TC (RTC). Moreover, the robust principal component analysis (RPCA) concept [41] has been used for RTC, resulting in the collaborative sparse and low-rank transforms model (CSLRT) [42] and robust tensor ring completion (RTRC) [43].

Although the ℓp-norm and its variants exhibit superior performance over the Frobenius norm in the presence of the outliers, the former is inferior to the latter in the Gaussian noise scenarios. Moreover, the ℓp-norm applies to whole tensor, implying that all the elements are considered as anomaly-contaminated. However, only a small proportion of tensors are typically corrupted by anomalies, and thus its performance is not optimal in impulsive noise. Conceptually, the RPCA model attains excellent performance in both the white Gaussian noise and gross error scenarios. Nevertheless, in practice, its solution depends on the choice of the regularization parameter. That is, for different data and noise, practitioners need to carefully adjust the penalty parameter to achieve good performance, which is time-consuming.

In this work, our aim is to devise an RTC method to suppress the white Gaussian noise and/or impulsive noise without the need to tune any parameter. We address RTC using the idea of the capped Frobenius norm, corresponding to a form of the truncated least-squares loss function [44], [45]. The interpretation of the upper bound of this norm is the boundary between the normal and outlier-contaminated entries. Therefore, we exploit the normalized median absolute deviation to update the bound, such that the capped Frobenius norm is capable of effectively resisting the white Gaussian noise and/or impulsive noise. We then combine the capped Frobenius norm with tensor ring decomposition to tackle RTC. Although the capped Frobenius norm is nonconvex, we use the half-quadratic theory [46], [47] to convert the resultant problem into a tractable form, that is, convex optimization with respect to (w.r.t) each individual variable. Subsequently, we adopt the proximal block coordinate descent (PBCD) method [48] to handle the multivariable optimization in which one block is updated while the remaining blocks are fixed at each iteration. Moreover, we establish the convergence of the proposed algorithm. Specifically, the suggested method ensures the objective function value to converge and the variable sequence to have a subsequence converging to a critical point. Our main contributions are summarized as follows.

1) We adopt the normalized median absolute deviation to obtain a robust standard deviation of the fitting error. Then, given a confidence interval, the upper bound is adaptively determined. Therefore, the capped Frobenius norm achieves excellent performance in the presence of Gaussian noise and/or impulsive noise.

2) We simplify the capped Frobenius norm based formulation into a Frobenius norm optimization with a regularization term. Then, PBCD is adopted as the solver, and all the subproblems have closed-form solutions.

3) We analyze the convergence behavior of the proposed algorithm. We prove that the objective function value is guaranteed to be convergent while the variable sequence contains a subsequence converging to a critical point.

4) Experimental results using real-world images and videos demonstrate that the devised approach is superior to popular robust methods in the presence of impulsive noise. In addition, without tuning any parameter, the performance of our method approaches that of the Frobenius-norm-based approach in the white Gaussian noise scenarios.

The remainder of this article is organized as follows. In Section II, we introduce notations and preliminaries. The proposed algorithm is presented in Section III. Besides, we analyze its convergence behavior and computational complexity. In Section IV, numerical examples are included to evaluate the devised method by comparing with several state-of-the-art algorithms. Finally, concluding remarks are given in Section V.

II. BACKGROUND

In this section, notations and basic definitions are provided, and relevant works are reviewed.

A. Notations

Scalars, vectors, matrices, and tensors are denoted by italic, bold lower case, bold upper case, and bold calligraphic letters, respectively. For example, \( \mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \) signifies an Nth-order tensor, and its \((l_1, l_2, \ldots, l_N)\) entry is denoted by \( x_{l_1, l_2, \ldots, l_N} \) or \( \mathbf{X}(l_1, l_2, \ldots, l_N) \). The mode-n unfolding of \( \mathbf{X} \) is represented by \( \mathbf{X}_{[p]} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \). Given a tensor \( \mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3} \), its Frobenius norm and ℓp-norm with \( p \in (0, 2) \) are \( \|\mathbf{X}\|_F = \left( \sum_{i=1}^{I_1} \sum_{j=1}^{I_2} \sum_{k=1}^{I_3} x_{i,j,k}^2 \right)^{1/2} \) and \( \|\mathbf{X}\|_p = \left( \sum_{i=1}^{I_1} \sum_{j=1}^{I_2} \sum_{k=1}^{I_3} x_{i,j,k}^p \right)^{1/p} \), respectively. Consider a matrix \( \mathbf{X} \in \mathbb{R}^{I_1 \times I_2} \), \( \mathbf{X}^T \) is the transpose of \( \mathbf{X} \), and \( \text{tr}(\mathbf{X}) = \sum_{i=1}^{\min(I_1,I_2)} x_{i,i} \) is the trace of \( \mathbf{X} \). The vectorization operator
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x \in \mathbb{R}^I$</td>
<td>scalar</td>
</tr>
<tr>
<td>$X \in \mathbb{R}^{l_1 \times l_2}$</td>
<td>$I$-th order tensor</td>
</tr>
<tr>
<td>$\mathbf{X} \in \mathbb{R}^{l_1 \times l_2 \times \cdots \times l_N}$</td>
<td>$N$-th order tensor</td>
</tr>
<tr>
<td>$\pi_{i_1, i_2, \cdots, i_N}$ or $\mathcal{X}(i_1, i_2, \cdots, i_N)$</td>
<td>vertical slice of $\mathbf{X}$</td>
</tr>
<tr>
<td>$\mathcal{X}(i_1, \cdot, \cdot)$</td>
<td>$r$-th lateral slice of $\mathbf{X}$</td>
</tr>
<tr>
<td>$\mathcal{X}[n] \in \mathbb{R}^{l_1 \times \cdots \times l_{n+1} \times l_{n+1} \cdots l_N}$</td>
<td>mode-$n$ unfolding of $\mathbf{X}$</td>
</tr>
<tr>
<td>$|x|<em>2 = \sqrt{\sum</em>{i=1}^{I} x_{i}^2}$</td>
<td>$\ell_2$-norm</td>
</tr>
<tr>
<td>$</td>
<td>\mathbf{X}</td>
</tr>
<tr>
<td>$</td>
<td>\mathbf{X}</td>
</tr>
<tr>
<td>$\text{tr}(\mathbf{X})$</td>
<td>trace of $\mathbf{X}$</td>
</tr>
<tr>
<td>$\mathbf{X}^T$</td>
<td>transpose of $\mathbf{X}$</td>
</tr>
<tr>
<td>$\mathbf{X}^{T_n}$</td>
<td>tensor permutation of $\mathbf{X}$</td>
</tr>
<tr>
<td>$\mathbf{X}(:, i, :)$</td>
<td>$i$-th column of $\mathbf{X}$</td>
</tr>
<tr>
<td>$\mathbf{X}(::, j, :)$</td>
<td>$j$-th lateral slice of $\mathbf{X}$</td>
</tr>
<tr>
<td>$\text{vec}(\mathbf{X}) = [x_{1,1}^T; x_{1,2}^T; \cdots; x_{l_1,1}^T]^T$</td>
<td>vectorization operator</td>
</tr>
<tr>
<td>$\text{mat}(\text{vec}(\mathbf{X})) = \mathbf{X}$</td>
<td>matricization operation</td>
</tr>
</tbody>
</table>

In this work, (3) is represented by $\mathbf{X} = h(\mathbf{Y}_1 \cdots \mathbf{Y}_N)$ for conciseness.

### C. Related Work

Consider a partially observed tensor $\mathbf{M}_\Omega \in \mathbb{R}^{l_1 \times \cdots \times l_N}$ where $\Omega \in \mathbb{R}^{l_1 \times \cdots \times l_N}$ is a binary tensor, consisting of 0 and 1. In $\Omega$, $\Omega_{i_1, \ldots, i_N} = 1$ signifies that the corresponding $m_{i_1, \ldots, i_N}$ is observed, while $\Omega_{i_1, \ldots, i_N} = 0$ indicates a missing entry, namely, $m_{i_1, \ldots, i_N} = 0$. Therefore, $\mathbf{M}_\Omega$ has the form of

$$\mathbf{M}_\Omega = \mathbf{M} \odot \Omega.$$  

As in matrix completion, TC can be formulated as a rank minimization problem

$$\min_{\mathbf{X}} \text{rank}(\mathbf{X}), \text{ s.t. } \mathbf{X} \odot \Omega = \mathbf{M}_\Omega$$

where $\mathbf{X} \in \mathbb{R}^{l_1 \times \cdots \times l_N}$. Since the rank function is nonconvex and discrete, Zhang et al. [50] propose using tensor nuclear norm instead of the rank function to solve (6), leading to

$$\min_{\mathbf{X}} \|\mathbf{X}\|_{\text{TNN}}, \text{ s.t. } \mathbf{X} \odot \Omega = \mathbf{M}_\Omega$$

where $\|\cdot\|_{\text{TNN}}$ denotes the tensor nuclear norm, defined in [50]. Since the tensor nuclear norm is defined based on the third-order tensors, its application is limited. To handle higher order tensors, Wang et al. [29] suggest exploiting tensor ring decomposition to formulate the tensor recovery problem, resulting in

$$\min_{\mathbf{Y}_{\text{new}, \ldots, \text{new}}} \|\mathbf{M}_\Omega - h(\mathbf{Y}_1 \cdots \mathbf{Y}_N) \odot \Omega\|_F^2.$$  

However, the above-mentioned methods are not robust to the outliers since they adopt the Frobenius norm for error minimization. Consider a linear regression problem based on the Frobenius norm

$$\min_{\mathbf{X}} \|\mathbf{A}\mathbf{X} - \mathbf{Y}\|_F^2 = \min_{\mathbf{X}} \sum_{i,j=1}^{l_1} \sum_{i,j=1}^{l_1} (a_{i,j}x_{i,j} - y_{i,j})^2.$$  

If $y_{i,j}$ with $i \in [1, l_1]$ and $j \in [1, l_2]$ is corrupted by an outlier, the ratio between the corresponding residual $|a_{i,j}x_{i,j} - y_{i,j}|^2$ and that of a normal entry is severely enlarged by the Frobenius norm. Then, to minimize the whole fitting error, $x_{i,j}$ will tend to reduce $(a_{i,j}x_{i,j} - y_{i,j})^2$. For instance, consider $\mathbf{A} = [1, 2; 3, 4], \mathbf{X} = [-0.05, -0.3; 0.075, 0.275], \text{ and } \mathbf{Y} = [0.1, 0.25; 0.15, 0.2]$. If $\mathbf{Y}$ is mixed with the white Gaussian noise, leading to $\tilde{\mathbf{Y}} = [0.0941, 0.2685; 0.1360, 0.2182]$, then $\tilde{\mathbf{X}} = \mathbf{A}^{-1}\tilde{\mathbf{Y}} = [-0.0521, -0.3189; 0.0731, 0.2937]$. However, if $y_{1,1}$ is corrupted by an anomaly, such that $\tilde{\mathbf{Y}} = [3.25; 0.15, 0.2]$, then $\tilde{\mathbf{X}} = [-5.85, -0.3; 4.425, 0.275]$. That is, the Frobenius-norm-based solution deviates much from the ground truth.

One of the prevailing methods to resist anomalies is to adopt the $\ell_p$-norm with $p \in (0, 2)$ as the loss function [39], leading to

$$\min_{\mathbf{X}} \|\mathbf{X}\|_p, \text{ s.t. } \mathbf{X} \odot \Omega - \mathbf{M}_\Omega \|_p \leq \delta$$

where $\delta > 0$ is a tolerance parameter to control the fitting error, and $\|\cdot\|_n$ is defined as the logarithmic norm; see [39].
Another strategy is to exploit RPCA, and the corresponding algorithms include tensor nuclear norm with total variation regularization (TNTV) [51] and transformed t-SVD (TTSVD) [52]. Specifically, TTSVD considers the following optimization problem:

$$\min_{\mathbf{X} \in \mathcal{S}} \| \mathbf{L}^\top \mathbf{TNN} + \mu \| \mathbf{S} \|_1$$

s.t. $\mathbf{L} + \mathbf{S} = \mathbf{X}, \mathbf{X} \otimes \Omega = \mathbf{M}_\Omega$ (11)

where $\mu > 0$ is the penalty parameter, and $\| \|_\mathbf{TNN}$ is the transformed tubal nuclear norm, as defined in [52]. Ideally, (11) is able to attain good performance in both the Gaussian noise and impulsive noise via tweaking $\mu$. However, it is time-consuming to tune $\mu$ to attain satisfactory performance for different data and noise environments.

III. PROPOSED ALGORITHM

In this section, we first present the suggested method. Then, we analyze its convergence behavior and computational complexity.

A. Algorithm Development

To restrain gross errors, we suggest solving tensor recovery using the capped Frobenius norm, defined as follows:

$$\| \mathbf{X} \|_{CF} = \sqrt{\sum_{i=1}^n \min_{\mathbf{X} \in \mathcal{S}, \mathbf{X} \otimes \Omega = \mathbf{M}_\Omega} (x^2 + \theta^2)}$$ (12)

where $\theta > 0$ is an upper bound to suppress anomalies. Note that the upper limit $\theta$ can be considered as the threshold for differentiating the normal and anomaly-contaminated elements. In this study, we adopt the normalized median absolute deviation for its adaptive determination, which is shortly introduced in (28).

Before proceeding, we highlight the advantages of the capped Frobenius norm over the Frobenius norm and $\ell_p$-norm with $p \in (0, 2)$. Compared with the Frobenius norm, in impulsive noise, the capped Frobenius norm is able to restrain the outliers. When there are no outliers, it reverts to the Frobenius norm. In comparison to the $\ell_p$-norm, the capped Frobenius norm resists gross errors, but it does not affect the normal entries. As a result, the capped Frobenius norm has a better performance than the $\ell_p$-norm in the presence of impulsive noise, as well as achieves comparable performance to the Frobenius norm in the Gaussian noise scenarios.

We then combine the capped Frobenius norm with tensor ring decomposition to formulate the RTC problem as follows:

$$\min_{\mathbf{Y} \in \mathcal{S}, \mathbf{Y} \otimes \Omega = \mathbf{M}_\Omega} \| \mathbf{M}_\Omega - h(\mathbf{Y}_1 \cdots \mathbf{Y}_N) \otimes \Omega \|_{CF}$$ (13)

where $\mathbf{Y}_n \in \mathbb{R}^{R \times L \times R}$ with $R$ being the predefined multilinear tensor ring rank. Since the capped Frobenius norm is nonconvex, (13) w.r.t. each variable is nonconvex. We exploit the half-quadratic theory to convert (13) into a tractable problem, such that it is a convex optimization w.r.t. each individual variable. The relevant background is introduced in the following lemma.

Lemma 1: [46]: Given $\phi(y)$ and $\psi(x)$, if $\phi(y)$ makes $f(y) = y^2 - \phi(y)$ convex, and $\psi(x)$ generates a convex function $g(x) = x^2 + \psi(x)$, then we have

$$\phi(y) = \inf_x ((y - x)^2 + \psi(x)), \quad y \in (-\infty, +\infty) \quad (14a)$$

$$\psi(x) = \sup_y ((-y - x)^2 + \phi(y)), \quad x \in (-\infty, +\infty). \quad (14b)$$

Based on Lemma 1, we derive an equivalent convex function of $\phi_\theta(y) = \min (y^2, \psi_\theta)$, as described in Theorem 1.

Theorem 1: Let $\phi_\theta(y) = \min (y^2, \theta^2)$, then minimizing $\phi_\theta(y)$ is equivalent to

$$\min_{x,y} (y - x)^2 + \psi_\theta(x)$$ (15)

where $\psi_\theta(x)$ is

$$\psi_\theta(x) = \begin{cases} -\theta^2, & |x| \leq \theta \\ \theta^2, & |x| > \theta. \end{cases}$$ (16)

The proof is provided in Appendix A. Note that all the appendices are presented in supplementary material.

Moreover, we introduce a set $\Phi$ involving the coordinates of the observed entries in $\mathbf{M}_\Omega$, defined as follows:

$$\Phi = \{(l_1, \ldots, l_N) | \Omega_{l_1,\ldots,l_N} = 1\}. \quad (17)$$

Based on $\Phi$, (13) is reexpressed as follows:

$$\min_{\mathbf{Y} \in \mathcal{S}} \| \mathbf{M}_\Omega - h(\mathbf{Y}_1 \cdots \mathbf{Y}_N) \otimes \Omega - \mathbf{S}_\Theta \|_F + \psi(\mathbf{S}_\Theta). \quad (18)$$

In accordance to Theorem 1, (18) is equivalent to

$$\min_{\mathbf{Y} \in \mathcal{S}} \| \mathbf{M}_\Omega - h(\mathbf{Y}_1 \cdots \mathbf{Y}_N) \otimes \Omega - \mathbf{S}_\Theta \|_F + \psi(\mathbf{S}_\Theta). \quad (19)$$

Defining $\psi_\Theta(\mathbf{S}_\Theta) = \sum_{(l_1,\ldots,l_N) \in \Phi} \psi_\Theta(\Omega_{l_1,\ldots,l_N})$, (19) is then reformulated as follows:

$$\min_{\mathbf{Y} \in \mathcal{S}} \| \mathbf{M}_\Omega - h(\mathbf{Y}_1 \cdots \mathbf{Y}_N) \otimes \Omega - \mathbf{S}_\Theta \|_F + \psi(\mathbf{S}_\Theta). \quad (20)$$

It is clear that (20) is a multivariable nonconvex optimization problem, but w.r.t. each individual variable, it is convex. To tackle (20), PBCD is adopted as the solver, resulting in the following iterative procedure:

$$\mathbf{Y}_n^{k+1} = \arg \min_{\mathbf{Y}_n} \| \mathbf{M}_\Omega - h(\mathbf{Y}_1^{k} \cdots \mathbf{Y}_N^{k}) \otimes \Omega - \mathbf{S}_\Theta \|_F^2$$

$$+ \psi_\Theta(\mathbf{S}_\Theta) \quad (21a)$$

$$\mathbf{Y}_1^{k+1} = \arg \min_{\mathbf{Y}_1} \| \mathbf{M}_\Omega - h(\mathbf{Y}_1^{k+1} \cdots \mathbf{Y}_N^{k}) \otimes \Omega - \mathbf{S}_\Theta^{k+1} \|_F^2$$

$$+ \lambda \| \mathbf{Y}_1 - \mathbf{Y}_1^{k} \|_F^2 \quad (21b)$$

$$\mathbf{Y}_2^{k+1} = \arg \min_{\mathbf{Y}_2} \| \mathbf{M}_\Omega - h(\mathbf{Y}_1^{k+1} \cdots \mathbf{Y}_N^{k}) \otimes \Omega - \mathbf{S}_\Theta^{k+1} \|_F^2$$

$$+ \lambda \| \mathbf{Y}_2 - \mathbf{Y}_2^{k} \|_F^2 \quad (21c)$$

$$\vdots$$

$$\mathbf{Y}_N^{k+1} = \arg \min_{\mathbf{Y}_N} \| \mathbf{M}_\Omega - h(\mathbf{Y}_1^{k+1} \cdots \mathbf{Y}_N^{k+1}) \otimes \Omega - \mathbf{S}_\Theta^{k+1} \|_F^2$$

$$+ \lambda \| \mathbf{Y}_N - \mathbf{Y}_N^{k} \|_F^2 \quad (21d)$$
where $\lambda > 0$ is the proximal parameter. It is seen that the PBCD alternatively updates one of the variables with fixing the $N$ remaining variables at each iteration.

We first tackle (21a). It is easy to know that its solution is only determined by the entries with $(l_1, \ldots, I_N) \in \Phi$. We thus reformulate (21a) as a vector optimization problem

$$s^{k+1} = \arg \min_{s} \|r^k - s\|_2^2 + \psi_0(s) \quad (22)$$

where $r^k \in \mathbb{R}^{[\Omega]}$ consists of the observed entries of $\mathcal{R}^k = \mathcal{M} - \mathcal{S}^k \cup \Omega$. The procedure for obtaining $r^k$ from $\mathcal{R}^k$ is illustrated with the use of a third-order tensor as follows. Given $\Omega \in \mathbb{R}^{3 \times 3 \times 2}$, such that

$$\Omega_{:,1} = [0 \ 0 \ 1 \ 0 \ 0 \ 1] \quad \Omega_{:,2} = [1 \ 0 \ 0 \ 1 \ 0 \ 0]. \quad (23a)$$

Then, $r^k = [r_{1,1,1}, r_{1,2,1}, r_{2,3,1}, r_{1,1,2}, r_{2,2,2}, r_{1,3,2}]$ with $r_{i,j,k}$ being the $(i, j, k)$ entry of $\mathcal{R}^k$.

We then derive the closed-form solution to (22), describing in the following lemma.

**Lemma 2:** For the following optimization problem:

$$s^{k+1} = \arg \min_{s} \phi(s) = \arg \min_{s} (r - s)^2 + \psi_0(s). \quad (24)$$

Its optimal solution is $T_{\psi}(r)$, defined as follows:

$$s^{k+1} = T_{\psi}(r) = \begin{cases} 0, & |r| < \theta \\ r, & |r| \geq \theta. \end{cases} \quad (25)$$

Besides, the subgradient of $\phi(s)$ at minimizer $s^{k+1}$ is

$$\partial \phi(s^{k+1}) = \begin{cases} 0 \in [- (r + \theta), \theta - r], & |r| < \theta \\ 0, & |r| \geq \theta. \end{cases} \quad (26)$$

The proof is provided in Appendix B.

In (22), $s^{k+1}$ only depends on $r^k$, and hence its optimal solution is

$$s^{k+1} = T_{\psi}(r^k). \quad (27)$$

Note that $s^{k+1}$ is affected by the parameter $\theta^k$. We suggest updating $\theta^k$ prior to computing $s^{k+1}$ for better performance. Since $r^k$ is defined as the fitting error at the $k$th iteration, if the mean of the fitting error is assumed $0$, $-\theta < r < \theta$ is considered as a confidence interval to identify anomalies. To guarantee the convergence, it requires $\theta^k$ to be nonincreasing

$$\theta^k = \min(\hat{\theta}, \theta^{k-1}) \quad (28)$$

where $\hat{\theta}$ is determined by a robust measure for standard deviation, namely, the normalized median absolute deviation method [33]

$$\hat{\theta} = \zeta \times 1.4826 \times \text{Med}(|r^k - \text{Med}(r^k)|). \quad (29)$$

Here, $\zeta > 0$ controls the confidence interval range, and Med() is the sample median operator. In Section IV, we will investigate the impact of $\zeta$ on the recovery performance in the Gaussian noise and impulsive noise scenarios.

After obtaining $s^{k+1}$, $s^{k+1}$ is updated via the inverse operation of constructing $r^k$ from $\mathcal{R}^k$.

We then handle (21b)–(21d). Since they have the same structure, we detail the derivation procedure for one of them, say $y^{k+1}$, without loss of generality. To simplify expressions, the optimization problem for updating $y^{k+1}$ is reexpressed as follows:

$$y_{n^{k+1}} = \arg \min_{\nu} \|g_{\mathcal{N}} - h(y_{1}^{k+1} \cdots y_{n-1}^{k+1}, y_{n}^{k+1} \cdots y_{N}^{k+1}) \circ \Omega_n^{(2)} \|_F^2 + \lambda \|y_n - y_n^{k}\|_F^2 \quad (30)$$

where $g_{\mathcal{N}} = \mathcal{M} - \mathcal{S}_{\mathcal{N}}^{k}$. Using tensor permutation operation, (30) is equivalent to

$$y_{n^{k+1}} = \arg \min_{\nu} \|g_{\mathcal{N}}^{T_n} - h(y_{1}^{k+1}, y_n^{k+1}) \circ \Omega_n^{T_n} \|_F^2 + \lambda \|y_n - y_n^{k}\|_F^2 \quad (31)$$

where $y = y_{1}^{k+1} \cdots y_{n-1}^{k+1} \cdots y_{N}^{k+1}$ with the dimensions of $R \times (I_1 \cdots I_{N-1} I_{N+1} \cdots I_N)$ $\times R$. We further adopt mode-$n$ unfolding to recast (31) as the following matrix optimization problem:

$$y_{n^{k+1}} = \arg \min_{\nu} \|g_{\mathcal{N}}^{T_n} - h(y_{1}^{k}, y_n^{k}) \circ \Omega_n^{T_n} \|_F^2 + \lambda \|y_n - y_n^{k}\|_F^2 \quad (32)$$

where the dimensions of $g_{\mathcal{N}}^{T_n}$, $h(y_{1}^{k}, y_n^{k})$, and $\Omega_n^{T_n}$ are $I_n \times (I_1 \cdots I_{N-1} I_{N+1} \cdots I_N)$. Since the $l$th lateral slice of $\mathcal{Y}_n$, denoted as $\mathcal{Y}_n(l, :)$, corresponds to the $l$th row of $h(y_{1}^{k}, y_n^{k})$, (32) can be split into $I_n$ subproblems

$$y_{n^{k+1}}(l, :) = \arg \min_{\nu} \|g_{\mathcal{N}}^{T_n}(l, :) - h(y_{1}^{k}, y_n^{k})(l, :) \circ \Omega_n^{T_n}(l, :) \|_F^2 + \lambda \|y_n(l, :) - y_n^{k}(l, :)\|_F^2 \quad (33)$$

where the lengths of $g_{\mathcal{N}}^{T_n}(l, :)$, $h(y_{1}^{k}, y_n^{k})(l, :)$, and $\Omega_n^{T_n}(l, :)$ are $I_1 \cdots I_{N-1} I_{N+1} \cdots I_N$. Equation (33) indicates that $\mathcal{Y}_n$ can be updated in a distributed or parallel manner. Similar to the update of $\mathcal{S}$, the solution to (33) is only affected by the observed entries, and thus (33) is equivalent to

$$y_{n^{k+1}}(l, :) = \arg \min_{\nu} \|g_{\mathcal{N}}(l, :) - h(y_{1}^{k}, y_n^{k})(l, :) \circ \Omega_n(l, :) \|_F^2 + \lambda \|y_n(l, :) - y_n^{k}(l, :)\|_F^2 \quad (34)$$

where $g_{\mathcal{N}} \in \mathbb{R}^{[\mathcal{N}]}$ and $h(y_{1}^{k}, y_n^{k}) \in \mathbb{R}^{[\mathcal{N}] \times [\mathcal{N}] \times \mathbb{R}}$ consist of the observed elements of $g_{\mathcal{N}}(l, :)$ and $\mathcal{Y}_n$, respectively. Herein, $\Omega_l$ only includes entries of 1 in $\Omega_n(l, :)$. We represent the tensor $\mathcal{Y}_n(l, :) \in \mathbb{R}^{[\mathcal{N}] \times \mathbb{R}}$ by a matrix $\mathcal{Y}_n \in \mathbb{R}^{[\mathcal{N}] \times \mathbb{R}}$. Based on the elementwise expression of tensor ring, (34) is reformulated as follows:

$$y_{n^{k+1}}(l, :) = \arg \min_{\nu} \sum_{j=1}^{I_n} \|g_{\mathcal{N}}(j, :) - h(y_{1}^{k}, y_n^{k})(l, :) \circ \Omega_n(l, :) \|_F^2 + \lambda \|y_n(l, :) - y_n^{k}(l, :)\|_F^2 \quad (35)$$

To handle (35), we introduce the following lemma.

**Lemma 3:** [29]: Consider $U \in \mathbb{R}^{I_1 \times I_2}$ and $V \in \mathbb{R}^{I_2 \times I_3}$, we have

$$\text{tr}(U \times V) = \text{vec}(V^T) \text{vec}(U). \quad (36)$$
Algorithm 1 CFN-RTC

Input: Partially observed tensor $\mathcal{M}_\Omega \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, binary tensor $\Omega \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, tensor ring rank $R$, maximum iteration number $K_{\text{max}}$, and proximal parameter $\lambda$.

Initialize: Randomize $\mathcal{Y}^0_n \in \mathbb{R}^{R \times I_a \times R}$ with $n \in [1, N]$ and $\theta^0 = 10$

for $k = 1, 2, \cdots, K_{\text{max}}$ do

1) Compute $\theta^k$ via (28)
2) Compute $s^k$ via (27)
3) Update $\mathcal{S}^k$ based on $s^k$

for $n = 1 : N$ do

1) Compute $y^k_n$ via (40)
2) Update $\mathcal{Y}^k_n$ based on $y^k_n$

end for

end for

Stop if stopping criterion is met.

Output: $\mathcal{X} = h(\mathcal{Y}^1, \mathcal{Y}^k, \cdots, \mathcal{Y}^k_N)

B. Comparison With RPCA

From (20), we see that the capped Frobenius-norm-based formulation is converted into a form similar to RPCA. The main difference lies on the regularization term, that is, the $\ell_1$-norm in prevailing RPCA and $\psi_\theta(\cdot)$ in our method. The convexity of the $\ell_1$-norm generates a tractable optimization; however, it may overpenalize large components, which causes the solution to deviate from the ground truth. Although our $\psi_\theta(\cdot)$ is nonconvex, the resultant subproblem is convex and has a closed-form solution. On the other hand, the performance of RPCA and our method are affected by an auxiliary parameter. To our best knowledge, the tradeoff parameter in RPCA requires tweaking manually to attain good performance. While for the proposed algorithm, $\theta$ is automatically updated using a robust statistics-based method.

C. Convergence Analysis

In this section, we analyze the convergence behavior of CFN-RTC. To facilitate presentation, the analysis is based on a third-order tensor, that is, $\mathcal{X} = h(\mathcal{Y}_1, \mathcal{Y}_2, \mathcal{Y}_3) \in \mathbb{R}^{I_1 \times I_2 \times I_3}$. It is worth mentioning that the analysis can be extended to higher order tensors. We first define the objective function value as follows:

$$L_{\theta^k}(\mathcal{S}^k_1, \mathcal{Y}^k_1, \mathcal{Y}^k_2, \mathcal{Y}^k_3) = \|\mathcal{M}_\Omega - h(\mathcal{Y}_1^k, \mathcal{Y}_2^k, \mathcal{Y}_3^k)\|_F + \psi_{\theta^k} (\mathcal{S}^k_1).$$

The convergence behavior of $L_{\theta^k}(\mathcal{S}^k_1, \mathcal{Y}^k_1, \mathcal{Y}^k_2, \mathcal{Y}^k_3)$ is provided in Theorem 2.

Theorem 2: Let $L_{\theta^k}(\mathcal{S}^k_1, \mathcal{Y}^k_1, \mathcal{Y}^k_2, \mathcal{Y}^k_3)$ be the objective function value generated by Algorithm 1, then we have the following statements.

1) $L_{\theta^k}(\mathcal{S}^k_1, \mathcal{Y}^k_1, \mathcal{Y}^k_2, \mathcal{Y}^k_3)$ is nonincreasing with all the variables’ update.
2) $L_{\theta^k}(\mathcal{S}^k_1, \mathcal{Y}^k_1, \mathcal{Y}^k_2, \mathcal{Y}^k_3)$ is lower bounded.

Therefore, $\{L_{\theta^k}(\mathcal{S}^k_1, \mathcal{Y}^k_1, \mathcal{Y}^k_2, \mathcal{Y}^k_3)\}_{k \in \mathbb{N}}$ is convergent.

The proof is provided in Appendix C.

We then analyze the sequence behavior in Theorem 3. The definition of the critical point is first introduced using the following lemma.

Lemma 4: [53]: Given a function $\phi(x)$, then $x^*$ is a critical point if $x^*$ meets one of the following statements.

1) $\nabla \phi(x^*) = 0$ in the case of smooth $\phi(x)$.
2) $0 \in \partial \phi(x^*)$ when $\partial \phi(x^*)$ is the subgradient with the nonsmooth $\phi(x)$.

Theorem 3: Let $\{(\mathcal{S}^k_1, \mathcal{Y}^k_1, \mathcal{Y}^k_2, \mathcal{Y}^k_3)\}$ be the sequence generated by Algorithm 1. For any initialization with finite $L_{\theta^k}(\mathcal{S}^k_1, \mathcal{Y}^k_1, \mathcal{Y}^k_2, \mathcal{Y}^k_3)$, $(\mathcal{S}^k_1, \mathcal{Y}^k_1, \mathcal{Y}^k_2, \mathcal{Y}^k_3)$ meets the following properties.

1) The sequence $(\mathcal{S}^k_1, \mathcal{Y}^k_1, \mathcal{Y}^k_2, \mathcal{Y}^k_3)$ is bounded.
2) There exists a subsequence $(\mathcal{S}^k_1, \mathcal{Y}^k_1, \mathcal{Y}^k_2, \mathcal{Y}^k_3)$ converging to an accumulation point $(\mathcal{S}^*, \mathcal{Y}^*_1, \mathcal{Y}^*_2, \mathcal{Y}^*_3)$.
3) The accumulation point $(\mathcal{S}^*, \mathcal{Y}^*_1, \mathcal{Y}^*_2, \mathcal{Y}^*_3)$ is a critical point.

The proof is provided in Appendix D.
missing data and 3-dB GMM noise. (a) Convergence of elements in $S$.

![Fig. 1. Convergence behavior of the objective function value with 50% randomly missing data and 3-dB GMM noise.](image)

For the update of $S$, the computational complexity is dominated by the calculation of $h(Y_1, Y_2, Y_3)$. One efficient method is to compute the observed entries, resulting in the complexity of $O(p I_1 I_2 I_3 R^3)$ where $p$ is the observation ratio. In addition, the complexity to update $\theta_k$ is $O(p I_1 I_2 I_3)$. For updating $Y^n_k(:, :, :)$, with $n = 1, 2, 3$ by (38), the complexity is $O(||\tilde{\Omega}|| R^4)$. Therefore, the complexity to update $Y^n_k$ is $O(p I_1 I_2 I_3 R^4)$ due to $p I_1 I_2 I_3 = \sum_{n=1}^{3} ||\tilde{\Omega}||$. As a result, the total computational complexity is $O(p I_1 I_2 I_3 R^4)$ per iteration.

### IV. EXPERIMENTAL RESULTS

In this section, we evaluate the CFN-RTC using synthetic data, real-world images, and videos. Note that $\lambda$ is set to $10^{-8}$ in all the experiments. The competing methods include tensor ring completion (TRC) [29], $\ell_p$-norm based tensor train completion ($\ell_p$-TTC) [37], LNOP [39], TNTV [51], TTSVD [52], and RTC with rank estimation (RTC-RE) [54].

#### A. Convergence Behavior

We first verify the convergence behavior of the suggested method based on a small-size synthetic data, i.e., $M \in \mathbb{R}^{10 \times 10 \times 10}$. The complete tensor is generated by three tensor ring factors, namely, $M_1 \in \mathbb{R}^{2 \times 10 \times 2}$, $M_2 \in \mathbb{R}^{2 \times 10 \times 2}$, and $M_3 \in \mathbb{R}^{2 \times 10 \times 2}$ whose entries obey the standard Gaussian distribution. Then, the incomplete noise-free tensor $M_0$ consists of randomly selected 50% entries of $M$. Moreover, $M_0$ is contaminated with independent impulsive noise which is modeled by a Gaussian mixture model (GMM). The probability density function (PDF) of GMM is given by the following equation:

$$p_n(v) = \frac{c_1}{\sqrt{2\pi \sigma_1^2}} \exp\left(-\frac{v^2}{2\sigma_1^2}\right) + \frac{c_2}{\sqrt{2\pi \sigma_2^2}} \exp\left(-\frac{v^2}{2\sigma_2^2}\right)$$  (44)

where $c_1 + c_2 = 1$ with $0 < c_1 < 1$, and $\sigma_1^2$ and $\sigma_2^2$ are the variances. To simulate the impulsive noise, we take $\sigma_1^2 > \sigma_2^2$ and $c_2 < c_1$. This means that sparse and high power noise

![Fig. 3. Scenery and its corrupted versions.](image)

![Fig. 4. PSNR versus $\zeta$.](image)
Fig. 5. Recovered Scenery images by different algorithms. The first and second rows contain the results with the random mask with impulsive noise and Gaussian noise, respectively. The third row shows the restored images with the deterministic mask and impulsive noise, while the fourth row contains the reconstructed pictures with deterministic mask and Gaussian noise.

Fig. 6. Average PSNR (APSNR) versus observation ratio.

samples corresponding to $\sigma_2^2$ and $c_2$ are mixed in Gaussian background noise with small variance $\sigma_1^2$. We set $\sigma_2^2 = 100\sigma_1^2$ and $c_2 = 0.1$. The signal-to-noise ratio (SNR) is defined as follows:

$$
\text{SNR} = \frac{\|M\|_{\ell_p}}{\|\Omega\|_{\ell_q} \sigma_0^2}
$$

where $\sigma_0^2 = \sum_{i=1}^2 c_i \sigma_i^2$ is the total noise variance.

The convergence of the objective function value is investigated in Fig. 1. It is seen that the objective function value is nonincreasing and converges within 20 iterations.

In addition, Fig. 2 depicts the sequence convergence behavior of four variables, namely, $Y_1 \in \mathbb{R}^{2 \times 10 \times 2}$, $Y_2 \in \mathbb{R}^{2 \times 10 \times 2}$, $Y_3 \in \mathbb{R}^{2 \times 10 \times 2}$, and $S \in \mathbb{R}^{10 \times 10 \times 10}$. It is noted that the number of curves for $S$ is much less than 1000 because of its sparsity. We have already proved analytically the subsequence convergence of $\{(S^i, Y_1^i, Y_2^i, Y_3^i)\}$, and the simulations corroborate our theory, showing that convergence happens within 20 iterations in this case.

Fig. 7. PSNR$^I$ versus outlier ratio.

Fig. 8. Eight images.
B. Image Inpainting

One popular application of TC is color image inpainting [55]. Color images involve RGB channels, and one channel can be modeled as a matrix. Therefore, color images can be represented by the third-order tensors. In practice, images may not be entirely acquired owing to the damage to the photosensitive device or shadow cast by other objects. Furthermore, images may be corrupted by the white Gaussian noise or impulsive noise during wireless transmission or bit errors in the signal acquisition stage. In the following experiments, we consider two types of noise, namely, strong Gaussian noise with $\sigma^2 = 0.01$ as well as impulsive noise generated by the mixture of weak white Gaussian noise with $\sigma^2 = 0.002$ and salt-and-pepper noise with $\tau = 0.2$ where $\sigma^2$ and $\tau$ are the variance and density coefficient, respectively.

The examined image is Scenery with dimensions of $256 \times 256 \times 3$ [56]. Besides, we investigate two types of masks, namely, random and fixed masks [57]. The random mask implies that the image has randomly distributed missing pixels, while the deterministic mask corresponds to regular stripes.

Fig. 3 depicts the original Scenery and four corrupted versions, i.e., random mask with Gaussian noise, random mask with impulsive noise, fixed mask with Gaussian noise, and fixed mask with impulsive noise. To evaluate recovery performance, two widely used metrics are adopted, namely, peak signal-to-noise ratio (PSNR) and structural similarity (SSIM). Note that large PSNR and SSIM indicate good restoration performance.

We first investigate the impact of $\zeta$ in (29) on recovery performance. The results are plotted in Fig. 4 where the incomplete image has 50% randomly missing pixels. We see that the PSNR with Gaussian noise increases with $\zeta$, while PSNR in impulsive noise scenarios, first increases and then reduces with boosting the value of $\zeta$. This is because a smaller $\zeta$ results in a narrow confidence interval, indicating that more entries are considered as the outlier-contaminated.

Table II shows the performance of different algorithms on eight images with random masks.

<table>
<thead>
<tr>
<th>Image</th>
<th>PSNR</th>
<th>SSIM</th>
<th>PSNR</th>
<th>SSIM</th>
<th>PSNR</th>
<th>SSIM</th>
<th>PSNR</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>28.06</td>
<td>0.8218</td>
<td>30.56</td>
<td>0.8288</td>
<td>32.07</td>
<td>0.8349</td>
<td>33.58</td>
<td>0.8410</td>
</tr>
<tr>
<td>2</td>
<td>26.49</td>
<td>0.7827</td>
<td>28.99</td>
<td>0.8039</td>
<td>30.50</td>
<td>0.8200</td>
<td>32.01</td>
<td>0.8361</td>
</tr>
<tr>
<td>3</td>
<td>24.90</td>
<td>0.7456</td>
<td>27.39</td>
<td>0.7667</td>
<td>28.89</td>
<td>0.7828</td>
<td>30.40</td>
<td>0.8039</td>
</tr>
<tr>
<td>4</td>
<td>23.31</td>
<td>0.7089</td>
<td>25.82</td>
<td>0.7290</td>
<td>27.33</td>
<td>0.7450</td>
<td>28.84</td>
<td>0.7667</td>
</tr>
</tbody>
</table>

Authorized licensed use limited to: SHENZHEN UNIVERSITY. Downloaded on August 30,2023 at 07:28:57 UTC from IEEE Xplore. Restrictions apply.
TABLE III

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Image 1</th>
<th>Image 2</th>
<th>Image 3</th>
<th>Image 4</th>
<th>Image 5</th>
<th>Image 6</th>
<th>Image 7</th>
<th>Image 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFN-RTC (Ours)</td>
<td>10.9</td>
<td>10.8</td>
<td>10.7</td>
<td>10.6</td>
<td>10.5</td>
<td>10.4</td>
<td>10.3</td>
<td>10.2</td>
</tr>
<tr>
<td>TRC [29]</td>
<td>10.3</td>
<td>10.2</td>
<td>10.1</td>
<td>10.0</td>
<td>9.9</td>
<td>9.8</td>
<td>9.7</td>
<td>9.6</td>
</tr>
</tbody>
</table>

elements. In the Gaussian noise scenarios, all the observed entries are not corrupted by anomalies, and thus a bigger $\zeta$ results in better recovery performance. Under impulsive noise, a small $\zeta$ leads to many entries to be mistaken as outliers, while a very large $\zeta$ cannot identify all the anomaly-contaminated entries. To achieve excellent performance in both types of noise, we select $\zeta = 3$ for the following experiments.

For the four observed images in Fig. 3, the restored pictures by CFN-RTC and its competitors are depicted in Fig. 5. The measurement metrics are listed below the corresponding recovered pictures. It is seen that the CFN-RTC attains the best performance on both the random and fixed masks in the presence of impulsive noise. In Gaussian noise, the performance of the CFN-RTC ranks second. It is worth pointing out that its performance is close to that of TRC adopting the Frobenius norm.

The effect of the percentage of missing pixels on performance is shown in Fig. 6 in which two types of noise are considered. The metric of APSNR signifies the average PSNR with Gaussian noise and impulsive noise. We see that the CFN-RTC attains better performance than TNTV, TRC, LNOP, TTSVD, $\ell_p$-TTC, and RTC-RE at all the observation ratios. Note that the recovery accuracy of the RTC-RE severely decreases when the missing percentage increases.

Moreover, we compare all the algorithms with different outlier ratios. The experimental results for random mask with 50% observed data are depicted in Fig. 7. It is seen that the CFN-RTC outperforms the existing algorithms no matter the outlier ratio is large or small.

Furthermore, eight well-known images, as shown in Fig. 8, are used to assess the inpainting performance. The results with 50% observation ratio for random mask are tabulated in Table II, while those of the deterministic mask are listed in Table III. It is seen that the CFN-RTC attains the highest PSNRs in the presence of impulsive noise, and better performance than TNTV, LNOP, TTSVD, $\ell_p$-TTC, and RTC-RE in the Gaussian noise scenarios. Therefore, the average performance of the CFN-RTC with two types of noise is superior to all the competitors. It can be known that the capped Frobenius norm is able to attain comparable performance to the Frobenius norm in normal situation. On the other hand, the runtimes of the proposed method are less than those of $\ell_p$-TTC and TRC-RE. Our approach and TRC are slower than LNOP, TTSVD, and TNTV since both of them adopt tensor ring decomposition. Although tensor ring factorization has a higher complexity than the t-SVD used by LNOP, TTSVD,
and TNTV, the former is able to handle higher order tensors, while the latter can only tackle the third-order tensors.

C. Video Restoration

The second application of TC is video restoration. Since LNOP, TTSVD, and TNTV only process the third-order tensors, we adopt grayscale videos to compare the CFN-RTC with the existing approaches. It is worth mentioning that the CFN-RTC is able to cope with higher order tensors. The examined dataset is YUV Video Sequences, and we select two typical ones, namely, Akiyo and Hall. The dimensions of each frame are $147 \times 176$. We use the first ten frames of both the videos to assess different algorithms, which is adopted in [39]. Thereby, the dimensions of each video are $147 \times 176 \times 10$. The recovery performance is evaluated using PSNR and SSIM.

Fig. 9 shows one of recovered frames of the Akiyo video under 50% randomly missing pixels. The first row shows the results with impulsive noise, while the second row depicts the restored frames with Gaussian noise where Gaussian and impulsive components are the same as the previous settings. We see that the CFN-RTC achieves higher PSNR and SSIM values than TRC, TNTV, LNOP, TTSVD, $\ell_p$-TTC, and RTC-RE in the presence of impulsive noise. In Gaussian noise, the CFN-RTC and TRC attain better performance than their competitors. However, the average performance of the CFN-RTC is the best among seven algorithms. Fig. 10 shows the plots of the average performance of all the frames. It is seen that the APSNRs of the CFN-RTC are larger than those of the other approaches.

Under the same condition as the Akiyo video, the results of the Hall video are shown in Figs. 11 and 12. Fig. 11 shows one of restored frames, while Fig. 12 shows the plots of the average performance of all the frames. It is seen that the CFN-RTC outperforms its competitors in impulsive noise. Although its performance ranks second in Gaussian noise, its average performance is superior to the competing algorithms in Fig. 12.

Furthermore, we investigate the recovery performance of different algorithms under a high percentage of missing pixels, namely, 80% randomly missing pixels. The experimental results are plotted in Figs. 13 and 14. It is seen that the suggested method still attains better performance than its competitors under a high missing percentage. Note that the performance of the RTC-RE severely degrades when the missing percentage increases.

---

1http://trace.eas.asu.edu/yuv/
V. Conclusion

In this article, we have devised an RTC algorithm using the capped Frobenius norm and tensor ring decompositions. The upper bound of the capped Frobenius norm is automatically updated using the normalized median absolute deviation strategy. The half-quadratic theory is used to simplify the nonconvex problem, resulting in a tractable task such that it becomes a convex optimization w.r.t. each individual variable. Then the PBCD method is exploited to handle the resultant problem, yielding an algorithm called CFN-RTC. The convergence behavior of the CFN-RTC is analyzed, that is, the objective function value is guaranteed to be convergent while the variable sequence has a subsequence to converge to a critical point. The experimental results on real-world images and videos demonstrate that the CFN-RTC achieves higher recovery accuracy than six popular algorithms in the presence of impulsive noise. Besides, its performance is comparable to the Frobenius-norm-based method without tweaking parameter in Gaussian noise.

REFERENCES


Zhi-Yong Wang was born in Henan, China. He received the B.S. degree in mechanical engineering from Zhengzhou University, Zhengzhou, China, in 2017, and the M.S. degree in mechanical engineering from Xi’an Jiaotong University, Xi’an, China, in 2020. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, City University of Hong Kong, SAR, China, supervised by Prof. Hing Cheung So.

His research interests include sparse recovery, robust signal processing, and low-rank approximation.

Zhang-Lei Shi received the Ph.D. degree from the Department of Electrical Engineering, City University of Hong Kong, Hong Kong, SAR, China, in 2021.

He is currently a Lecturer with the College of Science, China University of Petroleum (East China), Qingdao, China. His current research interests include neural networks and machine learning.

Hing Cheung So (Fellow, IEEE) was born in Hong Kong. He received the B.Eng. degree in electronic engineering from the City University of Hong Kong, Hong Kong, SAR, China, in 1990, and the Ph.D. degree in electronic engineering from The Chinese University of Hong Kong, Hong Kong, in 1995.

From 1990 to 1991, he was an Electronic Engineer with the Research and Development Division, Everex Systems Engineering Ltd., Hong Kong. From 1995 to 1996, he was a Post-Doctoral Fellow with The Chinese University of Hong Kong. From 1996 to 1999, he was a Research Assistant Professor with the Department of Electrical Engineering, City University of Hong Kong, where he is currently a Professor. His research interests include detection and estimation, multidimensional harmonic retrieval, robust signal processing, and sparse approximation.


Nicholas D. Sidiropoulos (Fellow, IEEE) received the Diploma degree in electrical engineering from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 1988, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland at College Park, College Park, MD, USA, in 1990 and 1992, respectively.

He is currently a Louis T. Rader Professor at the University of Virginia, Charlottesville, VA, USA. He has previously served on the faculty of the University of Minnesota, Minneapolis, MN, USA, and the Technical University of Crete, Kounoupitidiana, Greece. His research interests are in signal processing, communications, optimization, and tensor decomposition with applications in machine learning and communications.

Dr. Sidiropoulos is a fellow of EURASIP in 2014. He received the NSF/CAREER Award in 1998, the IEEE Signal Processing Society (SPS) Best Paper Award in 2001, 2007, 2011, and 2023, and the IEEE SPS Donald G. Fink Overview Paper Award in 2023. He received the IEEE Signal Processing Society Meritorious Service Award in 2010, the Distinguished Alumni Award of the Department of ECE, University of Maryland in 2013, the EURASIP Technical Achievement Award in 2022, and the IEEE SPS Claude Shannon—Harry Nyquist Technical Achievement Award in 2023. He served as the IEEE SPS Distinguished Lecturer (2008–2009), as the Vice President-Membership of IEEE SPS (2017–2019), and as the Chair for the SPS IEEE Fellow Evaluation Committee (2020–2021).